

REMARKS

Claims 1 - 32 are pending in the application. Claims 1 - 32 have been rejected. Claims 1, 14, 17, 18 and 20 have been amended. No new claims have been added.

Claims 1 and 3 - 17 and 22 - 32 stand rejected under 35 U.S.C. § 102(e) as being anticipated by Kosugi et al., U.S. Publication No. 20010044841 (Kosugi). Claims 2, 18, 19 and 21 stand rejected under 35 U.S.C. § 103(a) Kosugi, in view of Davis, U.S. Publication No. 20030070115 (Davis). These rejections are respectfully traversed.

The present invention generally relates to a method for accessing and writing event data to a log on failure of a computer system that is independent of the computer system's operating system. In the method, if it was determined that the BIOS program was not able to write the data to the event log, e.g., when the BIOS program encounters a subsequent failure such as an occurrence of a second critical event, then the system controller is configured to respond to the second failure by writing the data to the event log. The execution of the BIOS program may result in a second failure, thereby preventing the BIOS program from being completed.

More specifically, the present invention, as set forth by independent claim 1, relates to a method of accessing event data describing a failure. The method includes providing a computer system with a system management controller coupled between a processor bus and a local bus, configuring the system management controller to monitor a task of writing data to an event log, the task being executed by a Basic Input Output System (BIOS) program in response to the failure, monitoring the task for completion and accessing the event data if the task fails to complete and writing the event log via the system management controller in response to accessing the event data.

The present invention, as set forth by independent claim 14, relates to a method of accessing event data on a failure of a computer system. The method includes executing a BIOS program to access the event data in response to a first failure of the computer system, triggering a watchdog timer in a system management controller of the computer system, the watchdog timer being triggered substantially concurrent to the first failure,

the system management controller being coupled between a processor bus and a local bus, configuring the watchdog timer to allow the BIOS program to complete in absence of a second failure, determining whether the execution of the BIOS program caused the second failure, the second failure forcing the watchdog timer to expire, and the system management controller accessing the event data when the watchdog timer expires and writing the event log via the system management controller in response to accessing the event data.

The present invention, as set forth by independent claim 17, relates to a computer system which includes a processor, a memory coupled to the processor, a BIOS program stored in the memory, the BIOS program being operable to write data to an event log in response to a critical event and a system controller coupled to the memory and the processor and coupled between a processor bus and a local bus where the system controller is operable to receive an indication of the critical event and upon receipt of the indication, initiate operation of a timer, and determine whether the BIOS program has written the data to the event log within a configurable period of time defined by the timer and writing the data to the event log if the BIOS program has not written the data to the event log.

The present invention, as set forth by independent claim 18, relates to a method of responding to an event in a computer system having a processor and a system controller coupled between a processor bus and a local bus. The method includes issuing an interrupt to the processor in response to the event, detecting the interrupt at the system controller coupled to the processor, initiating a timer in the system controller upon detection of the interrupt, attempting to write data to an event log by executing a BIOS program, and the system controller determining whether the execution of the BIOS program resulted in writing data to the event log and writing data to the event log via the system management controller if the execution of the BIOS program did not result in writing data to the event log.

The present invention, as set forth by new independent claim 20, relates to a method for accessing and writing event data to a log on failure of a computer system

including a system management controller which is coupled between a processor bus and a local bus, the method includes monitoring a task of writing data to an event log via the system management controller, the task being executed by a Basic Input Output System (BIOS) program in response to the failure, monitoring the task for completion to determine whether the BIOS program was able to complete writing the data to the event log; accessing the event data if the task failed to complete and writing the event log via the system management controller if the task failed to complete.

Kosugi discloses a maintenance support apparatus and a maintenance method for automatically notifying an external remote maintenance system of the occurrence of a trouble at the time of starting a computer system until a system application is started. The maintenance support apparatus includes a server management support board and an integrated management panel which monitors a system. A start processing unit conducts start processing including a self diagnosis processing and a BOOT processing of an operating system and then starts an application when power of a computer system is turned on. A trouble notification unit controls the power of the computer system and integrally monitors a trouble of the start processing unit and a trouble during system operation. The trouble notification unit, provided as a server management support board, acquires log information stored in the start processing unit and notifies an external remote maintenance system of the log information as well as an alarm message through a dedicated network interface when the trouble notification unit detects the trouble of the start processing unit (system down).

The examiner cited to the following portion of Kosugi when setting forth that Kosugi discloses configuring a system management controller to monitor a task of writing data to an event log:

The server maintenance support board 36 notifies the *external remote maintenance server 16* of an alarm message indicating that system down occurs and the system log (BIOS log) at the time since the power of the system is turned on until the application is started using a network interface, i.e., if the server maintenance support board 36 is notified by the IMP panel board 34 of the system down caused by the trouble which occurs during a self diagnosis processing (POST diagnosis processing), a BOOT processing and an application start

processing conducted by the baseboard 30 by means of an electronic mail and an attachment file thereof. (Kosugi, Para. 29, emphasis added.)

The examiner cited to the following portion of Kosugi when setting forth that Kosugi discloses monitoring the task for completion and accessing the event data if the task fails to complete:

If the controller 88 is notified of the error during the BOOT processing, the controller 88 notifies *the remote maintenance server 16* of an alarm message indicating system down caused by the error which occurs during the BOOT processing of the baseboard 30 by means of an electronic mail through the interface 15 using the LAN module 46 serving as the network interface. Further, the controller 88 also acquires a system log stored in the nonvolatile memory 78 of the baseboard 30, to be specific, a BIOS log in addition to the alarm message indicating the system down during the BOOT processing, and notifies the remote maintenance server 16 of the system log as the attachment file of the electronic mail including the alarm message through the Internet 15. (Kosugi, Para 30, emphasis added.)

Kosugi does not disclose monitoring for the failure of a task to complete where the monitoring is via a system management controller that is included within the computer system in which the failure occurs.

Davis discloses a mechanism that allows remote diagnosis and management of a computer system, regardless of whether the computer system is installed as a stand alone unit or coupled to a computer network.

More specifically, Kosugi and Davis, taken alone or in combination, do not teach or suggest a method of accessing event data describing a failure where the method includes providing a computer system with a system management controller coupled between a processor bus and a local bus, configuring the system management controller to monitor a task of writing data to an event log, the task being executed by a Basic Input Output System (BIOS) program in response to the failure, monitoring the task for completion and accessing the event data if the task fails to complete and writing the event log via the system management controller in response to accessing the event data, all as required by claim 1. Accordingly, claim 1 is allowable over Kosugi and Davis. Claims 2 - 13 depend from claim 1 and are allowable for at least this reason.

Kosugi and Davis, taken alone or in combination, do not teach or suggest a method of accessing event data on a failure of a computer system where the method includes executing a BIOS program to access the event data in response to a first failure of the computer system, triggering a watchdog timer in a system management controller of the computer system, the watchdog timer being triggered substantially concurrent to the first failure, the system management controller being coupled between a processor bus and a local bus, configuring the watchdog timer to allow the BIOS program to complete in absence of a second failure, determining whether the execution of the BIOS program caused the second failure, the second failure forcing the watchdog timer to expire, and the system management controller accessing the event data when the watchdog timer expires and writing the event log via the system management controller in response to accessing the event data, all as required by claim 14. Accordingly, claim 14 is allowable over Kosugi and Davis. Claims 15 and 16 depend from claim 14 and are allowable for at least this reason.

Kosugi and Davis, taken alone or in combination, do not teach or suggest a *computer system which includes a system controller coupled between a processor bus and a local bus* where the system controller is operable to receive an indication of the critical event and upon receipt of the indication, initiate operation of a timer, and determine whether the BIOS program has written the data to the event log within a configurable period of time defined by the timer much less *write the data to the event log if the BIOS program has not written the data to the event log*, all as required by claim 17. Accordingly, claim 17 is allowable over Kosugi and Davis.

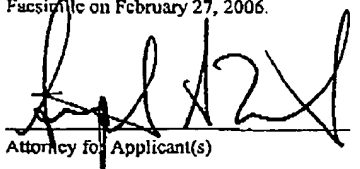
Kosugi and Davis, taken alone or in combination, do not teach or suggest a method of responding to an event in a computer system having a processor and *a system controller where the system controller is coupled between a processor bus and a local bus* where the method includes issuing an interrupt to the processor in response to the event, detecting the interrupt at the system controller coupled to the processor, initiating a timer in the system controller upon detection of the interrupt, attempting to write data to an event log by executing a BIOS program, and the system controller determining whether the execution of the BIOS program resulted in writing data to the event log and

writing data to the event log via the system management controller if the execution of the BIOS program did not result in writing data to the event log, all as required by claim 18. Accordingly, claim 18 is allowable over Kosugi and Davis. Claim 19 depends from claim 18 and is allowable for at least this reason.

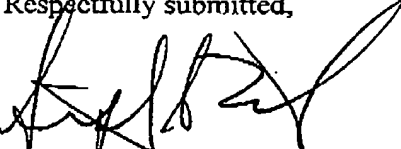
Kosugi and Davis, taken alone or in combination, do not teach or suggest a method for accessing and writing event data to a log on failure of a computer system including *a system management controller which is coupled between a processor bus and a local bus*, where the method includes monitoring a task of writing data to an event log via the system management controller, the task being executed by a Basic Input Output System (BIOS) program in response to the failure, monitoring the task for completion to determine whether the BIOS program was able to complete writing the data to the event log; accessing the event data if the task failed to complete and writing the event log via the system management controller if the task failed to complete, all as required by claim 20. Accordingly, claim 20 is allowable over Kosugi and Davis. Claims 21 - 32 depend from claim 20 and are allowable for at least this reason.

CONCLUSION

In view of the amendments and remarks set forth herein, the application is believed to be in condition for allowance and a notice to that effect is solicited. Nonetheless, should any issues remain that might be subject to resolution through a telephonic interview, the examiner is requested to telephone the undersigned, shown below at 512-338-9100.

I hereby certify that this correspondence is being sent to the COMMISSIONER FOR PATENTS via the USPTO Central Facsimile on February 27, 2006.	
	2/27/06
Attorney for Applicant(s)	Date

Respectfully submitted,


Stephen A. Terrile
Attorney for Applicant(s)
Reg. No. 32,946